University of Waterloo
CS240 Spring 2022
Assignment 5

Due Date: Wednesday, July 20 at 5:00pm

The integrity of the grade you receive in this course is very important to you and the University of Waterloo. As part of every assessment in this course you must read and sign an Academic Integrity Declaration before you start working on the assessment and submit it **before the deadline of July 20th** along with your answers to the assignment; i.e. **read, sign and submit A05-AID.txt now or as soon as possible**. The agreement will indicate what you must do to ensure the integrity of your grade. If you are having difficulties with the assignment, course staff are there to help (provided it isn't last minute).

The Academic Integrity Declaration must be signed and submitted on time or the assessment will not be marked.

Please read `https://student.cs.uwaterloo.ca/~cs240/s22/assignments.phtml#guidelines` for guidelines on submission. **Each question must be submitted individually to MarkUs as a PDF** with the corresponding file names: a5q1.pdf, a5q2.pdf, a5q3.pdf, a5q4.pdf, a5q5.pdf and a5q6.pdf. It is a good idea to submit questions as you go so you aren't trying to create several PDF files at the last minute.

Late Policy: Assignments are due at 5:00pm on Wednesday. Students are allowed to submit **one** late assignment, 2 days after the due date on Friday by 5:00pm. Assignments submitted after Friday at 5:00pm or Wednesday at 5:00pm (if you have already used your one late submission) will not be accepted for grading but may be reviewed (by request) for feedback purposes only. If you want to use your one time late assignment allowance, please send an email to "cs240@uwaterloo.ca" with the title "Using one time late assignment allowance".

Problem 1 Max-x in kd-trees [5 marks]

Give an algorithm to find the point with the maximum x-value in a 2D kd-tree, and analyze its complexity. For maximum credit, your algorithm must run in $o(n)$ time, where n is the number of points in the kd-tree. For simplicity, you may assume that n is a power of 4. If the run-time $T(n)$ of your algorithm satisfies a recurrence relation that has already been seen in class, you can take for granted the corresponding growth rate for $T(n)$, without giving the proof.

Problem 2 Huffman Coding [2+3 = 5 marks]

a) Trace the algorithm to build the Huffman trie for the string `mississauga`, and name the result T_1. Repeat the process for the string `massasauga`, and name the result T_2.

To break ties for tries of the same frequency, make the trie whose characters have a smaller ASCII sum the left child of the new root (that is, make it follow the "0" label).

b) Compare the compression rates of the tries from part (a) for encoding the string `massasagua`.

Problem 3 Run-Length Encoding [2+1+3+2 = 8 marks]

Suppose we want to encode a ternary string, where each symbol is 0, 1, or 2, to a ternary string using RLE.

a) What do you need to change in the RLE from the slides to make it work for a ternary string?

b) What are your ideas for making sure you are optimizing the compression rate?

c) Present your algorithm as a pseudocode.

d) What is the compression rate of your encoding scheme for the following ternary string?

012001122000111222000011112222000000000000001111111111111111222222222222222222.

Problem 4 Range Trees [3+4+4=11 marks]

a) Draw a 2-dimensional range tree of minimal height for the following set of average assignment and midterm grades:

$$\{(81, 70), (85, 68), (88, 88), (94, 68.5), (97, 75.5), (97.5, 100), (100, 80)\}$$

b) Assume that we have a set of n numbers (not necessarily integers) and we are interested only in the number of points that lie in a range rather than in reporting all of them. Describe how a 1-dimensional range tree (i.e., a balanced BST) can be modified such that a range counting query can be performed in $O(\log n)$ time (independent of s). Briefly justify that your algorithm is within the expected run-time.

c) Next, consider the 2-dimensional case where we have a set of n 2-dimensional points. Given a query rectangle R, we only want to find the number of points inside R, not the points themselves. Explain how to modify the Range Tree data structure discussed in class such that you can answer any of these counting queries in $O((\log n)^2))$ time. Briefly justify that your algorithm is within the expected run-time.

m	i	s	s	i	s	s	a	u	g	a	m	i	s	m	i	s	s	i	s	s	i	p	p	i	m	i	l	l

Table 1: Table for KMP problem (you may need to add rows, or delete some).

Problem 5 KMP [3+4=7 marks]

a) Compute the failure array for the pattern $P =$`ississi`.

b) Show how to search for pattern

$$P =\texttt{ississi}$$

in the text

$$T =\texttt{mississaugasmismississippimill}$$

using the KMP algorithm. Indicate in a table such as Table **??** which characters of P were compared with which characters of T. Follow the style in the example on slide 16 in Module 9.

- Place each character of P in the column of the compared-to character of T.
- Put brackets around the character if an actual comparison was not performed.
- Use a new row when sliding the pattern.
- You may not need all space in the table.

Problem 6 Boyer-Moore [1+3+3+3+5=15 marks]

a) Construct the last occurrence function L for pattern $P = ramammam$ where $\Sigma = a, k, m, r$.

c	a	k	m	r
L(c)				

b) Trace the search for P in $T = ramaamamkmamramammmam$ using the Boyer-Moore algorithm. Indicate in a table such as Table **??** which characters of P were compared with which characters of T. Follow the example on slides 23-24 in Module 9.

- Place each character of P in the column of the compared-to character of T.
- Put brackets around the character if they are known to match from the previous step (similar to the examples in the slides).
- Use a new row when sliding the pattern.
- You may not need all space in the table.

r	a	m	a	a	m	a	m	k	m	a	m	r	a	m	a	m	m	a	m

Table 2: Table for Boyer-Moore problem.

c) For any $m \geq 1$ and any $n \geq m$, give a pattern P and a text T such that the Boyer-Moore algorithm looks at exactly $\Theta(n/m)$ characters. Justify your answer.

d) For any $m \geq 1$ and any $n \geq m$ that is a multiple of m, give a pattern P and a text T such that the Boyer-Moore algorithm looks at all characters of the text at least once and returns with failure. Justify your answer.

e) A number of heuristics can be used with Boyer-Moore to reduce the number of comparisons performed between P and T. Suppose we use Boyer-Moore with only the Peek heuristic. The Peek heuristic states that if $P[j] \neq T[i]$ and $P[j-1] \neq T[i-1]$ then the next location to search for P at is $T[i+m-1]$. Show that the Peek heuristic may fail to find P in T, i.e., find a pattern P, and a text T containing P, such that Peek fails to find P in T. Justify your answer in a short paragraph.